Open vSwitch Acceleration

Contents of this whitepaper:

Virtualizing the Accelerated, Accelerating the Virtualized

Open vSwitch (OVS) fully deserves its reputation of being one of the most interesting and important open source projects. In order to abstract from the physical network infrastructure, OVS is widely used in datacenters to steer traffic among virtualized appliances running as virtual machines (VMs), apply access and security policies, and realize overlay networks by means of protocol tunneling.

As one might expect, such an important functional component of the network becomes critical from the performance perspective as well. While the original OVS implementations suffered from fairly bad throughput, recent changes have improved this significantly. Introduction of Microflow and Megaflow caches reduces the need to traverse the full match table chain. Optional DPDK interface bypasses Linux kernel networking stack, further improving overall system throughput. It has been shown that DPDK itself (not with OVS) is capable of 100G+ packet receive and transmit (see the record).

The world of application-specific network hardware accelerators is a completely different animal. Since the acceleration functions (such as cryptography, DPI, or even simple hardware packet filtering) are not generally available to the VMs through the OVS, the applications that make use of hardware acceleration are typically built as standalone hardware boxes. No advantages of virtualization, such as multitenancy, easy repurposing, or workload scaling, are typically available in the hardware acceleration domain.

Vendors and users are forced to make a decision: Use the flexible and future-proof virtualized platform with OVS, but with limited performance and no access to hardware acceleration, or run in the old-fashioned, less flexible non-virtual mode at high speed, with the use of various hardware accelerators.

Bringing them closer

Let's have a look at the technologies that will allow to bridge the gap between these two worlds. First of all, SR-IOV allows to map portions of the hardware directly to VMs. This extension to the PCI Express specification defines PCI Virtual Functions (VFs) that the CPU's IOMMU maps directly to the address space of the VM, without involvement of the hypervisor beyond initial configuration. If the network adaptor supports SR-IOV, it is (at least in theory) fairly straightforward to support dedicated per-VM acceleration modules. Each VM sees its associated VF with its accelerators, each VM-VF pair being logically isolated from the others. Read more about SR-IOV here.

What remains to be solved is steering the packets from the network to the VMs - the domain of OVS. Now the OVS cannot be used as it is, because we want the packets to pass through the VF accelerators before even reaching the CPU. In other words, packet switching functionality must occur *before* the acceleration functionality (in RX, of course). The solution here is to offload the OVS to the network adaptor as well. Due to the very good design of OVS, Microflow and Megaflow caches can be offloaded to the hardware. Most packets are matched in one of those hardware caches and sent directly to the appropriate VF accelerator, while the software OVS, that still runs in the hypervisor, handles only cache misses that occur for a small percentage of packets. See Fig. 1 for the conceptual scheme of virtualization-enabled hardware acceleration platform.

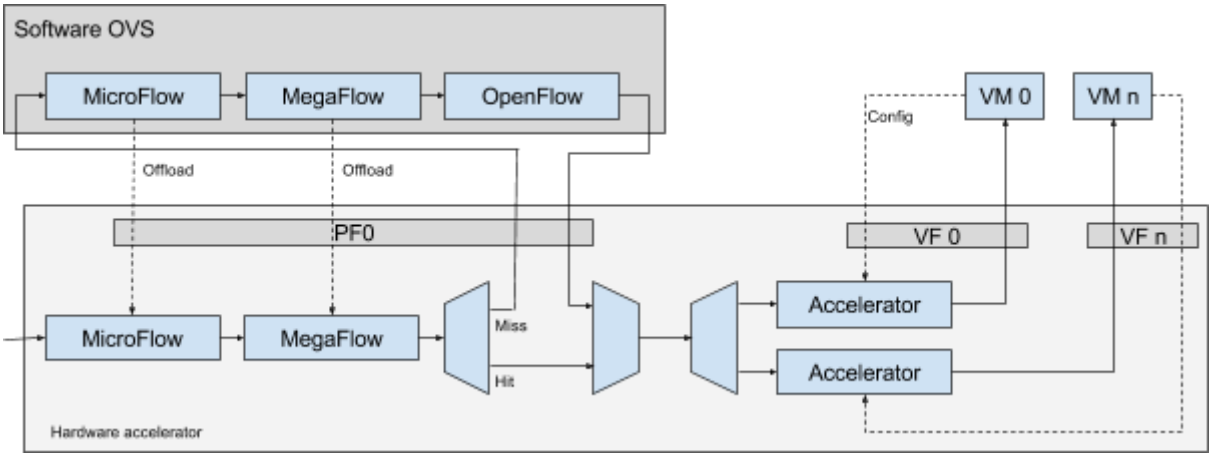

Fig. 1: Scheme of virtualization platform with OVS offload and hardware accelerators.

This setup has several nice features:

- OVS is accelerated by offloading Microflow and Megaflow caches to the hardware.
- Application hardware acceleration can be used in a virtualized environment.
- The system still has the OVS interface for configuration of network management and engineering - it is compatible with the existing cloud orchestration platforms.

Since the network adaptor gets more complicated, suitable hardware platform should be used for this task. While dedicated ASICs may offer the best throughput/Watt ratio, their inflexibility and long time to market is a serious disadvantage. Current FPGAs offer enough processing power to handle OVS offload as well as application specific hardware accelerators at 100 Gbps. Moreover, the direct structural programmability of FPGAs ensures predictable (and often constant!) latency and throughput. Finally, an FPGA-based accelerator will easily survive even large changes in functionality:

- New network protocols
- New tunneling mechanisms
- Different cache/match table structure
- Different application acceleration functions available to VMs

So you may run a virtualized DPI applications with pattern matching hardware offload and VXLAN tunneling on Monday, but switch to VLAN-based distribution of traffic among virtualized VPN terminators using cryptographic function offload on Tuesday. And you may add your custom secret sauce protocol support to that mix on Wednesday, if you want.

But who will take care of the dreadful FPGA firmware programming? There is one missing piece: the P4. This high level, platform agnostic and domain specific language provides a way to describe functionality of networking

devices. Protocol support, matching table structure, packet modifications, even external functions (accelerators!) are easily described in P4. P4 is a perfect match for OVS and custom functions acceleration. P4-generated FPGA firmware comes with API (read Whitepaper), so that it is easily integrated into software systems such as OVS. With P4, the data center's network engineer defines how exactly the OVS (or other packet processing task) is offloaded to the hardware, while the application engineer chooses appropriate hardware acceleration for the virtualized application.

Solved by Netcope

Netcope Technologies has all the building blocks necessary to create the described platform for acceleration of virtualized functions (or a platform for virtualization of accelerators, if you will):

- Netcope FPGA Boards are FPGA-based programmable network interface cards covering all the latest network technologies: 100G, 40G, and 10G Ethernet. The newest family member, NFB-200G2QL is a PCIe card mounted with the latest Xilinx Virtex UltraScale+ FPGA to process 2x100G Ethernet traffic at wire speed. It supports up to 256 SR-IOV Virtual Functions and 200Gbps throughput to host system, which, together with a small mechanical footprint (low-profile, half-length), makes it feasible for data center/virtualization workloads
- Netcope Development Kit is a toolset for rapid development of custom hardware-accelerated network applications based on Netcope FPGA Boards. It is based on a sophisticated build system and a collection of IP cores and software. It offers comprehensive environment that enables prototyping of an application in the shortest time possible, which is an invaluable feature for solution vendors, integrators and R&D teams. Netcope's record-breaking 100+ Gbps DPDK modules are included in NDK (read more). SR-IOV support is handled in NDK as well.
- Netcope P4 to VHDL compiler automatically generates VHDL source code from P4 sources, together with the API for management of the generated circuit.
- Netcope cryptography and pattern matching IP cores are examples of application specific accelerators. Other (custom or 3rd party) accelerators can be added to NDK easily.